rev. 6/14/00

NUCLEAR ENGINEERING EDUCATION RESEARCH
(NEER) GRANT PROGRAM

Grant Number DE-FG07-99ID13771

DEVELOPMENT OF A METHOD FOR QUANTIFYING THE RELIABILITY OF NUCLEAR SAFETY-RELATED SOFTWARE

ANNUAL REPORT

July 1, 2000

Submitted by

Prof. Michael W. Golay
Department of Nuclear Engineering
Massachusetts Institute of Technology
Cambridge MA 02139

INTRODUCTION

The work of our project is intended to help in introducing digital technologies into nuclear power plant safety related software applications. Success can likely yield substantial improvements in both nuclear power safety and economics. However, the obstacles to showing that new digital systems are sufficiently reliable have proven to be greater than had initially been appreciated, and has denied nuclear power plants the benefits of modern informatic technologies. Consequently, nuclear power plants today are digital dinosaurs with poor prospects for using digital technologies in the future.

In our project we are working to show how one can utilize a combination of modern software engineering methods: design process discipline and feedback, formal methods, automated computer aided software engineering tools, automatic code generation, and extensive feasible structure flowpath testing to improve software quality. These elements are used to promote complete system performance requirements capture, translation of these into the corresponding software specifications with production of the corresponding code; and systematic, extensive testing of the resulting software. A more detailed treatment of how this can be done is presented in our project proposal.

The tactics for achieving this goal include reducing the introduction of defects into software, ensuring that the software structure is kept simple, permitting routine testing during design development, permitting extensive finished product testing in the input data space of most likely service and using test-based Bayesian updating to estimate the probability that a random software input will encounter an error upon execution. From the results obtained the software reliability can be both improved and its value estimated. Hopefully our success in the project's work can aid the transition of the nuclear enterprise into the modern informatic world.

In our work we are utilizing three safety-related, nuclear power plant instrumentation and control software examples for implementation and refinement of the project's methods. These examples are obtained from ABB-Combustion Engineering (renamed Westinghouse Electric Company Nuclear Systems since May 2000), and our work is being performed with their collaboration. The initial software example is that of the digital Signal Validation Algorithm (SVA), which is used for selecting the plant instrumentation signal set which is to be used as the input to the digital Plant Protection System (PPS). This is the system which automatically trips the reactor when it receives appropriate signals.

The Signal Validation Algorithm sweeps all of the signals of interest and through a voting logic decides which signals can be trusted sufficiently to be used by the Plant Protection System. Each sweep is repeated 60 time per second, and the sequence of signals examined is the same in each sweep.

ACTIVITIES TO-DATE

Our work to-date has been concerned with initiating the project and working through the steps of software development and testing outlined above, using the example, Signal Validation

Algorithm program of ABB-Combustion Engineering. The elements of our efforts are the following:

1. We acquired the ABB-Combustion Engineering Signal Validation Algorithm program and got it running
2. We acquired the Signal Validation Algorithm specifications
3. We encoded the Signal Validation Algorithm specifications into the 001 tool suite system and generated results concerning faults in the specifications
4. We obtained the Signal Validation Algorithm testing input set used by ABB-Combustion Engineering
5. We purchased an adequate computer and got the 001 tool suite operating on it
6. We commenced Signal Validation Algorithm testing using the project's computer.

1. Acquiring the Signal Validation Algorithm program and Executing it on a Personal Computer (PC) [Proposal Tasks 2 and 3]

At the beginning of the project we acquired the Signal Validation Algorithm program from ABB-Combustion Engineering, and executed it using a PC. Doing this required considerable effort, as the Signal Validation Algorithm was developed for use with the ABB-Combustion Engineering System 80+ nuclear power plant design. The Signal Validation Algorithm was developed but has not been deployed as ABB-Combustion Engineering has not yet sold a plant of this design. Consequently the records and expertise concerned with the program have not been maintained since its development. Thus, obtaining the correct version of the program took longer than would have been the case with a program in current use.

The Signal Validation Algorithm has gone through the full ABB-Combustion Engineering testing program and is ready for commercial deployment, but it has not yet been used in the field. Consequently, the field use feedback which might be used to discover defects not identified in the pre-deployment phase of work has not yet come into play. For this reason this initial software example is especially interesting for our project as it is at the stage of development as would be software for which our software reliability estimation method is intended.

2. Acquiring the Signal Validation Algorithm Specifications [Proposal Task 1]

A key element of modern software development is detecting software defects in the software specifications, as typically about half of those identified in fully developed code are found to originate in the specifications. Thus, we have begun our investigation of the Signal Validation Algorithm at that level. In order to do this we have obtained the specifications of the Signal Validation Algorithm, and have utilized the 001 computer assisted software engineering (CASE) tool for mapping its variable and functional hierarchies (Ref. 1).

3. Encoding the Signal Validation Algorithm Specifications in the 001 Tool Suite System and Generating Specifications Testing Results [Proposal Task 1]

The purpose of mapping a program's variable and functional hierarchies is to identify inconsistencies and incompleteness within the specifications. The 001 tool suite was developed for this purpose and the project team have used it in previous software reliability-related work. Once the program structure implied in the specifications has been captured one also obtains a statement of the logical structure of the program from which the coped of the program itself can be generated, and which can provide a basis of testing whether the program implicit in the specifications can actually provide the intended results.

From the variable and functional hierarchy maps of the specifications we were able to identify several defects of inconsistency, ambiguity and incompleteness within the original SVA specification. These defects had been discovered previously in the ABB-Combustion Engineering Signal Validation Algorithm development and testing program. Thus, our effort in this phase of the work was primarily confirmatory, to ensure that the more systematic and automated approach of our project could identify defects previously identified via more informal, manually based efforts. The Signal Validation Algorithm code which we received for use in our project reflected corrections in the specifications identified in that previous effort.

We have since corrected the Signal Validation Algorithm specifications to remove the defects which we identified at the specifications testing stage of our work.

4. Obtaining the Signal Validation Algorithm Executable Flowpath Testing Input Set Used by ABB-Combustion Engineering [Proposal Tasks 3 and 4]

As a basis for our software testing work we have obtained the ABB-Combustion Engineering testing input set for the SVA. We are now in the process of investigating the degree to which it interrogates all of the executable paths of the Signal Validation Algorithm. Doing this is important in our work, as a key concept in obtaining high software reliability is taking advantage of the relative simplicity of nuclear safety-related software in order to achieve a much larger coverage of the set of executable paths within the software than is typically feasible in conventional examples of software testing and quality verification.

5. Purchasing an Adequate Computer and Utilizing the 001 Tool Suite on that System [Proposal Task 1]

In the early stages of our work we utilized a work station owned by Hamilton Technologies Inc. for running the 001 Tool Suite, and also for becoming trained in use of the tool. During 2000 we have acquired a computer for our own use and have installed the 001 Tool Suite on it. Doing this proved to be somewhat time consuming as faulty hardware in the purchased computer introduced frustrating delays into our work.

6. Commencing Signal Validation Algorithm Executable Flowpath Testing Using the Project's Computer [Proposal Tasks 3 and 4]

Since May 2000 we have been able to use the 001 Tool Suite in the project's computer, and we have commenced the testing program of the Signal Validation Algorithm. That testing remains in progress. Among the issues to be addressed during testing are those of formulating the testing program. We are beginning from the testing program developed at ABB-Combustion Engineering, but we expect to need to make it more elaborate. Our plan for Bayesian up-dating of the probability of encountering errors in using the software as software errors are found during testing is described elsewhere (Ref. 2), and we shall follow it during the Signal Validation Algorithm current testing phase.

Work to-date is summarized in a paper submitted for the 2000 Winter Meeting of the American Nuclear Society.

FUTURE WORK

During the coming year we anticipate completion shortly of the Signal Validation Algorithm testing program, revision of our research approach to reflect lessons learned during the Signal Validation Algorithm-related exercise, and initiation of work on a second software example (to be decided jointly with ABB-Combustion Engineering, but likely to be the Plant Protection System algorithm).

REFERENCES

1. M.H. Hamilton, *Increasing Quality and Productivity with a "Development Before the Fact" Paradigm*, Hamilton Technology Inc. Document, 1992.
2. Y. Sui and M.W. Golay, "Baysian Estimation of the Number of Errors in a Computer Program and its Relation to Software Reliability," *Probabilistic Safety Assessment and Management (PSAM 4)*, A. Mosleh and R.A. Bari, eds., Vol. 2, Springer-Verlag London Ltd., 1998.